CLAIMS

We claim:

1. A method, comprising:

receiving programming instructions to execute one or more shared resource threads (shreds) via an instruction set architecture (ISA);

configuring one or more instruction sequencers via the ISA; and

executing the one or more shreds simultaneously with a microprocessor that includes multiple instruction sequencers.

2. The method of claim 1, wherein the ISA provides a private state for each shred of the one or more shreds, and wherein the private state is associated with one of a plurality of registers including general purpose registers, floating point registers, MMX registers, segment registers, a flags register, an instruction pointer, control and status registers, SSE registers, and a MXCSR register.

3. The method of claim 1, further comprising sharing a state among a plurality of shreds of the one or more shreds, while maintaining the state privately among one shred of the plurality of shreds, wherein the state shared is associated with one of a plurality of registers including a control register, a flags register, memory management registers, a local descriptor table register, a task register, debug registers, model specific registers, shared registers, and shred control registers.

4. The method of claim 1, further comprising:

sharing a state among a plurality of shreds of the one or more shreds; and

storing the state in one or more registers.

5. The method of claim 1, wherein the one or more shreds share a current privilege level and share a common address translation.

6. The method of claim 1, wherein the ISA includes instructions to create, and destroy the one or more shreds.

7. The method of claim 1, further comprising communicating between the one or more shreds via one or more shared registers.

8. The method of claim 1, further comprising sharing a system state among the one or more shreds.

9. The method of claim 1, wherein the one or more shreds determine remote and local relationships to each shred of the one or more shreds.

10. The method of claim 1, further comprising associating the one or more shreds with a thread;

wherein an application program controls the one or more shreds directly, including scheduling of the one or more shreds, and

wherein an operating system executed by the multiprocessor schedules one or more threads.

11. The method of claim 1, further comprising:

associating the one or more shreds with a thread; and

suspending the one or more shreds belonging to the thread when a context switch request is received through a single shred of the one or more shreds.

12. The method of claim 11, further comprising:

storing one or more shred states associated with the one or more shreds when the context switch request is received.

13. The method of claim 1, further comprising:

a first process reporting one or more exceptions; and

a second process servicing the one or more exceptions.

14. The method of claim 13, further comprising:

receiving the one or more exceptions from an application program; and

determining whether to report the one or more exceptions to the operating system.

15. The method of claim 14, further comprising:

prioritized-reporting of the one or more exceptions and one or more context changes to the operating system; comprising

receiving the one or more exceptions simultaneously via different shreds of the one or more shreds; and

servicing one of the one or more exceptions and the one or more context changes according to the prioritized-reporting while suspending exception processing of other exceptions of the one or more exceptions.

16. The method of claim 1, wherein the one or more shreds perform input/output (I/O) functions and computation functions.

17. An apparatus, comprising:

a microprocessor that includes multiple instruction sequencers; and

a plurality of user-level multithreading registers coupled to the microprocessor that

receives programming instructions to execute one or more shreds in accordance with an instruction set architecture (ISA);

configures one or more instruction sequencers via the ISA; and

executes the one or more shreds simultaneously.

18. The apparatus of claim 17, wherein the plurality of user-level multithreading registers further comprises a plurality of shared shred registers to facilitate communication between a plurality of shreds and to facilitate synchronization between the plurality of shreds.

19. The apparatus of claim 18, wherein the plurality of user-level multithreading registers further comprises a plurality of shred control registers to manage the plurality of shreds.

20. The apparatus of claim 17, wherein the plurality of user-level multithreading registers comprises a first register that enables an operating system or BIOS to enable multithreading architecture extensions for user-level multithreading.

21. The apparatus of claim 17, wherein the ISA allows a private state for each shred of the one or more shreds, and wherein the private state is associated with one of a plurality of registers including general purpose registers, floating point registers, MMX registers, segment registers, a flags register, an instruction pointer, control and status registers, SSE registers, and a MXCSR register.

22. The apparatus of claim 17, wherein the ISA allows sharing a state among a plurality of shreds of the one or more shreds, while maintaining the state privately among one shred of the plurality of shreds, wherein the state shared is associated with one of a plurality of registers including a control register, a flags register, memory management registers, a local descriptor table register, a task register, debug registers, model specific registers, shared registers, and shred control registers.

23. The apparatus of claim 17, further comprising:
sharing a state among a plurality of shreds of the one or more shreds; and
storing the state in the plurality of user-level multithreading registers.

24. The apparatus of claim 17, wherein the one or more shreds share a current privilege level and share a common address translation.

25. The apparatus of claim 17, wherein the ISA includes instructions to create, and destroy the one or more shreds.

26. The apparatus of claim 17, further comprising communicating between the one or more shreds via the plurality of user-level multithreading registers.

27. The apparatus of claim 17, further comprising sharing a system state among the one or more shreds.

28. The apparatus of claim 17, wherein the one or more shreds determine remote and local relationships to each shred of the one or more shreds.

29. The apparatus of claim 17,

wherein an application program controls the one or more shreds directly, including scheduling of the one or more shreds, and

wherein an operating system executed by the multiprocessor schedules one or more threads.

30. The apparatus of claim 17, wherein the ISA associates the one or more shreds with a thread; and the processor suspends the one or more shreds when a context switch request is received through a single shred of the one or more shreds.

31. The apparatus of claim 30, wherein the microprocessor stores one or more shred states associated with the one or more shreds when the context switch request is received.

32. The apparatus of claim 17, wherein the ISA:

reports one or more exceptions by a first process; and

services the one or more exceptions by a second process.

33. The apparatus of claim 32, wherein the microprocessor:

receiving the one or more exceptions from an application program; and

determines whether to report the one or more exceptions to the operating system.

34. The apparatus of claim 33, wherein the ISA

permits prioritized reporting of the one or more exceptions and one or more context changes to the operating system upon receiving the one or more exceptions simultaneously via different shreds of the one or more shreds; and

services one of the one or more exceptions and the one or more context changes according to the prioritized-reporting while suspending exception processing of other exceptions of the one or more exceptions.

35. An article of manufacture, comprising:

a machine-accessible medium including data that, when accessed by a machine, cause the machine to perform operations comprising,

receiving programming instructions to execute one or more shared resource threads (shreds) via an instruction set architecture (ISA);

configuring one or more instruction sequencers via the ISA; and

executing the one or more shreds simultaneously with a microprocessor that includes multiple instruction sequencers.

36. The article of manufacture of claim 35, wherein the ISA provides a private state for each shred of the one or more shreds, and wherein the private state is associated with one of a plurality of registers including general purpose registers,

floating point registers, MMX registers, segment registers, a flags register, an instruction pointer, control and status registers, SSE registers, and a MXCSR register.

37. The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising sharing a state among a plurality of shreds of the one or more shreds, while maintaining the state privately among one shred of the plurality of shreds, wherein the state shared is associated with one of a plurality of registers including a control register, a flags register, memory management registers, a local descriptor table register, a task register, debug registers, model specific registers, shared registers, and shred control registers.

38. The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising sharing a state among a plurality of shreds of the one or more shreds; and storing the state in one or more registers.

39. The article of manufacture of claim 35, wherein the one or more shreds share a current privilege level and share a common address translation.

40. The article of manufacture of claim 35, wherein the ISA includes instructions to create, and destroy the one or more shreds.

41. The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising communicating between the one or more shreds via one or more shared registers.

42. The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising sharing a system state among the one or more shreds.

43. The article of manufacture of claim 35, wherein the one or more shreds determine remote and local relationships to each shred of the one or more shreds.

44. The article of manufacture of claim 35,

wherein an application program controls the one or more shreds directly, including scheduling of the one or more shreds, and

wherein an operating system executed by the multiprocessor schedules one or more threads.

45. The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising:

associating the one or more shreds with a thread; and

suspending the one or more shreds belonging to the thread when a context switch request is received through a single shred of the one or more shreds.

46. The article of manufacture of claim 45, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising:

storing one or more shred states associated with the one or more shreds when the context switch request is received.

47. The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising:

a first process reporting one or more exceptions; and

a second process servicing the one or more exceptions.

48. The article of manufacture of claim 47, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising:

receiving the one or more exceptions from an application program; and

determining whether to report the one or more exceptions to the operating system.

49. The article of manufacture of claim 48, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising:

prioritized-reporting of the one or more exceptions and one or more context changes to the operating system; comprising

receiving the one or more exceptions simultaneously via different shreds of the one or more shreds; and

servicing one of the one or more exceptions and the one or more context changes according to the prioritized-reporting while suspending exception processing of other exceptions of the one or more exceptions.

50. The article of manufacture of claim 35, wherein the one or more shreds perform input/output (I/O) functions and computation functions.

51. A system, comprising:

a microprocessor;

a plurality of user-level multithreading registers coupled to the microprocessor, wherein the plurality of user-level multithreading registers comprises a first register that enables an operating system or BIOS to enable multithreading architecture extensions for user-level multithreading; and

an instruction set architecture (ISA) for a 64-bit architecture or 32-bit architecture compatible with the microprocessor and the plurality of user-level multithreading registers.

52. The system of claim 51, wherein the plurality of user-level multithreading registers further comprises a plurality of shared shred registers to facilitate communication between a plurality of shreds and to facilitate synchronization between the plurality of shreds.

53. The system of claim 52, wherein the plurality of user-level multithreading registers further comprises a plurality of shred control registers to manage the plurality of shreds.

54. The system of claim 53, wherein the microprocessor:

receives programming instructions to execute one or more shreds in accordance with the ISA;

configures one or more instruction sequencers via the ISA; and

executes the one or more shreds simultaneously.

55. A system, comprising:

a microprocessor, including

a plurality of user-level multithreading registers coupled to the microprocessor; and

memory coupled to the microprocessor that stores an instruction set architecture (ISA) compatible with the microprocessor and the plurality of user-level multithreading registers, wherein the memory is from a plurality of memory devices including DRAM, flash, and EEPROM,

wherein the plurality of user-level multithreading registers and the ISA enable multithreading architecture extensions for user-level multithreading.

56. The system of claim 55, wherein the plurality of user-level multithreading registers further comprises a plurality of shared shred registers to facilitate communication between a plurality of shreds and to facilitate synchronization between the plurality of shreds.

57. The system of claim 56, wherein the plurality of user-level multithreading registers further comprises a plurality of shred control registers to manage the plurality of shreds.

58. The system of claim 57, wherein the microprocessor:

receives programming instructions to execute one or more shreds in accordance with the ISA;

configures one or more instruction sequencers via the ISA; and

executes the one or more shreds simultaneously.